European Symposium on Computer Aided Process Engineering – 15
L. Puigjaner and A. Espuña (Editors)

A software toolbox for the dynamic optimization of nonlinear processes

María-Sonia G. García[a], Eva Balsa-Canto[b], Antonio A. Alonso[a] and Julio R. Banga[a, *]

[a] Process Engineering Group, IIM-CSIC (Spanish Council for Scientific Research)
C/Eduardo Cabello 6, 36208 Vigo (SPAIN)

[b] Dept. of Applied Mathematics II
Universidad de Vigo, 36200 Vigo (SPAIN)

Abstract

This contribution describes the development and implementation of a novel software toolbox, NDOT, for the dynamic optimization (open loop optimal control) of nonlinear processes. This modular and flexible toolbox combines the control vector parameterization approach with a number of local and global nonlinear programming solvers and suitable dynamic simulation methods. NDOT is able to solve dynamic optimization problems for both lumped and distributed nonlinear processes. Its performance (robustness and efficiency) is illustrated considering a representative set of nonlinear (lumped and distributed) benchmark problems.

Keywords: dynamic optimization, optimal control, distributed parameter systems, control vector parameterization, parametric sensitivities

1. Introduction

The dynamic optimization problem (DO), also called open loop optimal control, considers the computation of time dependent operating conditions (controls) so as to improve a certain index (e.g. reduce production costs, improve product quality, meet safety requirements and environmental regulations, etc.). This problem can be mathematically formulated as follows:

Find $\mathbf{u}(t)$ along $t \in [t_0, t_f]$ which minimize (or maximize) a performance index J:

$$J = \phi(\mathbf{x}(\xi, t_f), \mathbf{y}(t_f), t_f) + \int_{t_0}^{t_f} L\left(\mathbf{x}(\xi, t), \mathbf{y}(t), \mathbf{u}(t), t\right) dt \tag{1}$$

subject to:

- the system dynamics that for the general case of distributed parameter systems (DPSs) corresponds to a set of partial and ordinary differential equations (PDAEs):

$$F\left(\mathbf{x}, \mathbf{x}_\xi, \mathbf{x}_{\xi\xi}, \mathbf{x}_t, \dot{\mathbf{y}}, \mathbf{y}, \mathbf{u}, t\right) = 0 \tag{2}$$

with the corresponding initial and boundary conditions.

- the bounds for the control variables:

[*] Corresponding author, email: julio@iim.csic.es

$$\mathbf{u}^L \le \mathbf{u}(t) \le \mathbf{u}^U \quad (3)$$

- and possibly other process constraints (path, point and/or final time):

$$\mathbf{c}\left(\mathbf{x}(\xi,t),\mathbf{y}(t),\mathbf{u}(t),t\right) \le 0 \quad (4)$$

where ξ are the spatial variables, t the time, $\mathbf{x}(\xi,t)$ are the state variables depending on both time and position (not present for lumped processes), $\mathbf{y}(t)$ the state variables depending only on time, $\mathbf{u}(t)$ is the vector of control variables and t_f the final time.

In practice, the DO of distributed parameter systems involves transforming the original PDAEs, Eqn. (2), into an equivalent lumped system, and then applying lumped-system DO methods. In this regard, the numerical method of lines (NMOL, Schiesser, 1991) and the finite element method (FEM) are the most popular techniques for spatial discretization, offering solutions to a wide variety of problems and geometries. The evolution of the field is then described by a large-scale, and possibly stiff, set of ordinary differential equations (ODEs) whose solution requires an adequate initial value problem (IVP) solver.

The solution of the resulting DO problem is a challenging task. Although there are several numerical alternatives available, the control vector parameterization (CVP) method seems to be the most convenient for large scale ODE systems and has been selected as the basis for this work. CVP proceeds dividing the control variables into a number of elements (ρ) and approximating each element with a low order polynomial. The polynomial parameters become the decision variables in an outer nonlinear programming problem (NLP) whose solution may be approached using a standard NLP solver. Note that the evaluation of the objective function and the first and second order information required by most of the local and global NLP solvers, involves the solution of an extended IVP which includes the system dynamics and the corresponding parametric sensitivities, as described in Balsa-Canto et al. (2001, 2004).

2. NDOT toolbox features

2.1. Main characteristics

NDOT (Nonlinear Dynamic Optimization Toolbox) has been implemented as a Matlab (www.mathworks.com) toolbox. NDOT is based on the CVP scheme, so the solution of the original dynamic optimization problem is approximated by solving a main NLP with an inner IVP problem. NDOT is particularly useful regarding the following issues:

- it can automatically handle high level definitions of nonlinear distributed processes. These systems are discretized and solved via suitable interfaces to finite element (FEM) or method of lines (MOL) packages.
- it includes the possibility of symbolically deriving first and (projected) second order parametric sensitivities (as recently presented by Balsa-Canto et al., 2002). The solution of the extended IVP provides gradient and projected Hessian information to be used together with suitable local NLP solvers.
- it offers global optimization (GO) solvers for the main NLP (which is frequently nonconvex), thus ensuring the proper solution of a number of challenging problems. The solution of these GO solvers can be later refined with any of the local NLP solvers also included.

NDOT has been designed with modularity and flexibility in mind, allowing the user to choose the NLP and IVP solvers from an ample set of state of the art options. Moreover, NDOT is easily extendible, i.e. interested users can add new NLP and/or IVP solvers. For the solution of the outer NLP problem, NDOT currently offers the following options: `FMINCON` (Anonymous, 2003), `SNOPT`, `NPSOL`, `MINOS`, `CONSOLVE`, `NLPSOLVE` and `OQNLP` (as implemented in Tomlab; Holmström et al., 2004), `FSQP` (Zhou et al., 1997) and `SOLNP` (Ye, 1989). Currently, only `OQNLP` offers global optimization, but other global solvers are being implemented. These solvers are called in a suitable way to exploit the first (and, sometimes, second) order information coming from sensitivities. With respect to the solution of the inner IVPs, NDOT currently offers all `ode*` integrators of Matlab, plus `LSODE` and `LSODES`. We are currently developing gateways to several state of the art IVP solvers.

2.2. Components and other characteristics

The NDOT main component is the so-called NDOTpp (pre-processor), which is written in Matlab, and it uses the Symbolic Manipulation Toolbox. Given a problem definition, by means of simple user-defined input (a simple Matlab data structure, as in Table 1), NDOTpp generates Matlab code for the drivers of the NLP and IVP solvers, plus the necessary code for the sensitivities. These drivers are also automatically interfaced with existing Dynamic Link Libraries for the selected NLP and IVP solvers.

After this pre-processing, the execution of the resulting code provides the user with detailed results (organized in data structures and, optionally, figures and tables). We have paid particular attention to computational efficiency, so the most costly procedures (e.g. IVP solution) can be automatically implemented in compiled f90/f77 code if needed. It should be noted that all this is done without requiring any intervention from the user.

Table 1. Input data for the dynamic optimization of a problem with NDOT.

%% Example: van der Pol	
Prob.name = 'vpol'	%% Name of problem
Prob.nf = 3	%% Number of states
Prob.nu = 1	%% Number of controls
Prob.J = 'y(3)'	%% Performance index
Prob.tf = 5	%% Final time
Prob.IVP.resid(1) = {'ydot(1)-(1-y(2)^2)*y(1)+y(2)-u(1)'}	%% System dynamics
Prob.IVP.resid(2) = {'ydot(2)-y(1)'}	
Prob.IVP.resid(3) = {'ydot(3)-y(1)^2-y(2)^2-u(1)^2'}	
Prob.IVP.IC = [0 1 0]	%% Initial conditions
Prob.IVP.Tol = 10^-5	%% Relative integration tolerance
Prob.IVP.Solver = 'ODE_45'	%% IVP solver
Prob.Control.u_lb = -0.3	%% Control lower bound
Prob.Control.u_ub = 1	%% Control upper bound
Prob.Control.u00 = 0.7	%% Control initial guess
Prob.Control.rho = 10	%% Control discretization
Prob.Opt.Tol = 10^-3	%% Rel. optimization tolerance
Prob.Opt.Solver = 'FMINCON'	%% Optimization solver

3. Case studies

In order to evaluate the performance of NDOT, the solution of a collection of 16 case studies is presented here. These cases have been selected to study the behaviour of NDOT with problems of different levels of complexity: lumped cases with up to four control variables, some of them presenting singular arcs or bang-bang control profiles; lumped cases with path and/or final time constraints; and, finally two, cases involving distributed parameter processes. These problems are listed using codenames in Table 2 and 3, with references to publications offering detailed statements. The results presented in Table 2 were obtained using a control discretization level ρ=10, with LSODE as the IVP solver. All the currently available NLP solvers were used, but only the best results are presented. The integration and optimization tolerances were 10^{-5} and 10^{-3}.

Table 2. Summary of results obtained with NDOT for the DO of lumped processes.

Case study	Opt.Solver	Results		Best known J
		J	Tcpu/s [a]	
Vpol [b]	OQNLP	2.92604	0.9	2.86728 [d]
Mixcatal [b, c]	SNOPT	0.048013	0.7	0.048069 [i]
Nishida [b, c]	SNOPT	1.67159	1.3	1.00891 [d]
Park_R_a [d]	SNOPT	92.8477	24.3	93.553 [d]
Park_R_b [e]	SNOPT	32.1159	2.7	32.691 [d]
Park_R_c [e]	OQNLP	32.1836	2.9	32.480 [d]
Non_diff [f]	NPSOL	58.7757	0.8	58.18 [f]
CSTR_a [e, c]	SNOPT	20.0895	6.4	20.0955 [d]
CSTR_b [e, c]	SNOPT	21.7046	11.9	21.8071 [d]
Lee_R_a [e]	OQNLP	6.15168	8.7	6.16 [j]
Lee_R_b [e]	NLPSOLVE	5.75874	5.6	5.77 [j]
Lee_R_c [e]	SNOPT	5.57044	4.5	5.58 [j]
Plugflow [g]	OQNLP	0.675815	12.9	0.67727 [g]
Penicillin [f, h]	SNOPT	87.7340	46.6	87.95 [f]

[a] Using a Pentium IV 2.4 GHz PC (431 Mflops according to the Linpack-100 benchmark); [b] Vassiliadis et al., 1999; [c] Opt.Tol=$1\cdot10^{-5}$ & Ivp.Tol=$1\cdot10^{-7}$; [d] Balsa-Canto, 2001; [e] Balsa-Canto et al., 2001; [f] Banga & Seider, 1996; [g] Carrasco & Banga, 1998; [h] For t_f=132 hours; [i] Dolán & Moré, 2001; [j] Tholudur & Ramírez, 1997.

Despite the somewhat coarse discretization (ρ=10) used for the controls, the results achieved are in good agreement with the best ones published in the literature. Further, they were obtained at a very modest computational cost, indicating the satisfactory efficiency of the current version of NDOT. Note, however, that more refined control discretizations would usually lead to better solutions, although increasing the computational effort. To illustrate this point, Figure 1 shows refined optimal control profiles for the two cases and the corresponding optimal performance indexes.

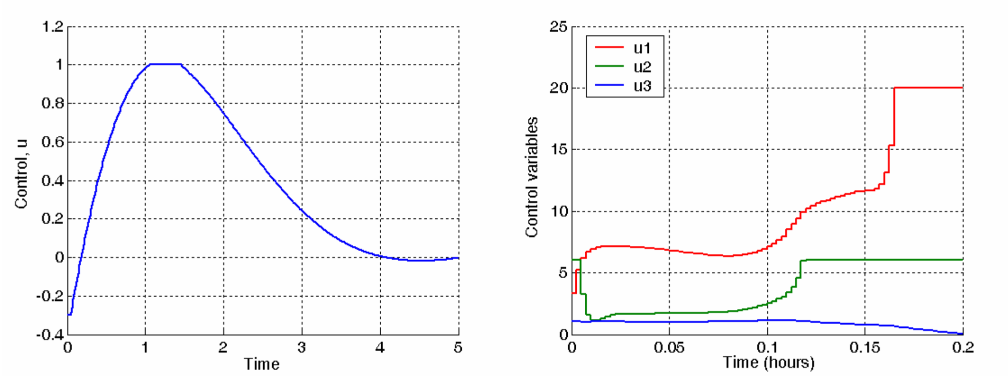

Figure 1. Optimal control profiles for Vpol (ρ=500, J=2.86229) and CSTR_a (ρ=80, J=20.0957)

Table 3 presents the results obtained for the dynamic optimization of two distributed parameter cases. The use of both NMOL and FEM are illustrated for the solution of one of the cases. LSODES was used as the IVP solver, allowing an efficient management of the sparsity of the ODE system Jacobian. Figure 2 shows the corresponding optimal control profiles.

Table 3. Summary of results obtained with NDOT for the DO of two distributed systems.

Case study & Reference	Discretization / PDE Solver	Results	
		J	Tcpu / s [a]
Slab [Balsa-Canto et al., 2004] [b]	NMOL / NPSOL	$1.87 \cdot 10^{-6}$	15.8
Slab [Balsa-Canto et al., 2004] [c]	FEM / SNOPT	$2.65 \cdot 10^{-6}$	18.7
Membrane [Balsa-Canto et al., 2004] [d]	NMOL / SNOPT	0.9338	112

[a] Using a Pentium IV 2.4 GHz PC (431 Mflops according to the Linpack-100 benchmark).
[b] Results for spatial discretization : 10. [c] Spatial discretization: 16. [d] Spatial discretization : 7.

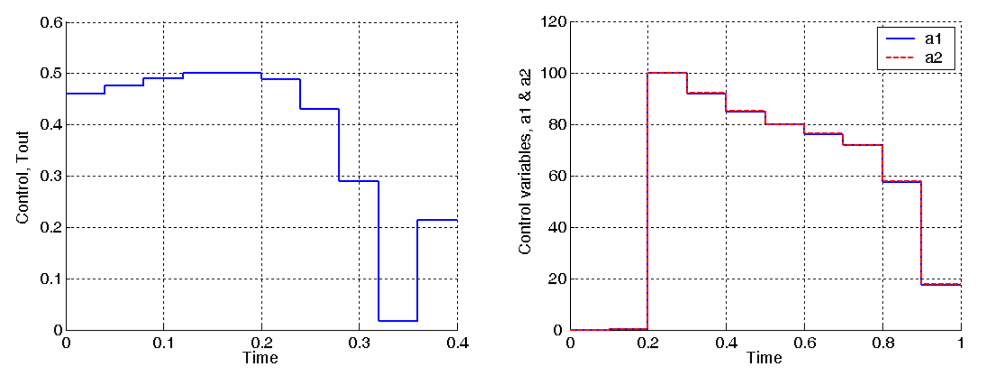

Figure 2. Optimal control profiles for the Slab and Membrane case studies (Table 3)

4. Conclusions

Here, we have presented NDOT, a Matlab toolbox for the dynamic optimization of nonlinear (lumped and distributed) processes. NDOT is based on the CVP scheme, and it has been designed with modularity and flexibility in mind. Currently, it already offers a broad set of efficient NLP and IVP solvers. Its excellent performance was illustrated considering a set of challenging dynamic optimization problems taken from the open literature.

References

Anonymous, 2003, User's Guide Optimization Toolbox for version 2.3. The Math Works, Inc.

Balsa-Canto, E., J. R. Banga, A. A. Alonso and V. S. Vassiliadis, 2004, Dynamic optimization of distributed parameter systems using second-order directional derivatives. Ind. Eng. Chem. Res., 43, 6756.

Balsa-Canto, E., V. S. Vassiliadis, A. A. Alonso and J. R. Banga, 2002, Restricted second order information for the solution of optimal control problems using control vector parameterization. J. Process Control 12(ER2), 243.

Balsa-Canto, E., J. R. Banga, A. A. Alonso and V. S. Vassiliadis, 2001, Dynamic optimization of chemical and biochemical processes using restricted second-order information. Computers & Chemical Engineering, 25, 4-6, 539.

Balsa-Canto, E., 2001, Algoritmos eficientes para la optimización dinámica de procesos distribuidos, PhD Thesis, University of Vigo, Spain (in Spanish).

Banga, J. R. and W. D. Seider, 1996, Global Optimization of Chemical Processes using Stochastic Algorithms. In "State of the Art in Global Optimization", Floudas C. A. and P. M. Pardalos (Eds), 563. Kluwer Academic Pub., Dordrecht, The Netherlands.

Carrasco, E. F. and J. R. Banga, 1998, A hybrid method for the optimal control of chemical processes. IEE Conf. Publ., 455, 2, 925.

Dolán, L. and J. Moré, 2001, Benchmarking optimization software with performance profiles. Argonne National Laboratory – ANL /MCS – P861-1200.

Holmström, K., A. O. Göran and M. M. Edwall, 2004, User's guide for Tomlab 4.3. www.tomlab.biz

Mekarapiruk W. and Luus R., 1997, Optimal control of inequality state constrained systems. Ind. Eng. Chem. Res., 36, 1686.

Schiesser, W. E., 1991, The Numerical Method of Lines: Integration of Partial Differential Equations. Academic Press, Inc. San Diego.

Tholudur, A. and W.F. Ramírez , 1997, Obtaining smoother singular arc policies using a modified Iterative Dynamic Programming algorithm. International Journal of Control, 68, 5, 1115.

Vassiliadis, V. S., E. Balsa-Canto, J. R. Banga, 1999, Second-order sensitivities of general dynamic systems with application to optimal control problems. Chem. Eng. Sci., 54, 17, 3851.

Ye, Y., 1989, SOLNP Users' Guide: a nonlinear optimization Program in MATLAB.

Zhou, J. L., A. L. Tits and C. T. Lawrence, 1997, User's Guide for FSQP version 3.7.

Acknowledgements

We thank the Spanish Ministry of Science and Technology (project AGL2001-2610-C02-02) and Xunta de Galicia (PGIDIT02PXIC40211PN) for financial support. Author Mª S. G. García gratefully acknowledges financial support (FPI fellowship) from the Spanish Ministry of Science and Technology.